VARIABLE WIDTH BLOCK CIPHER

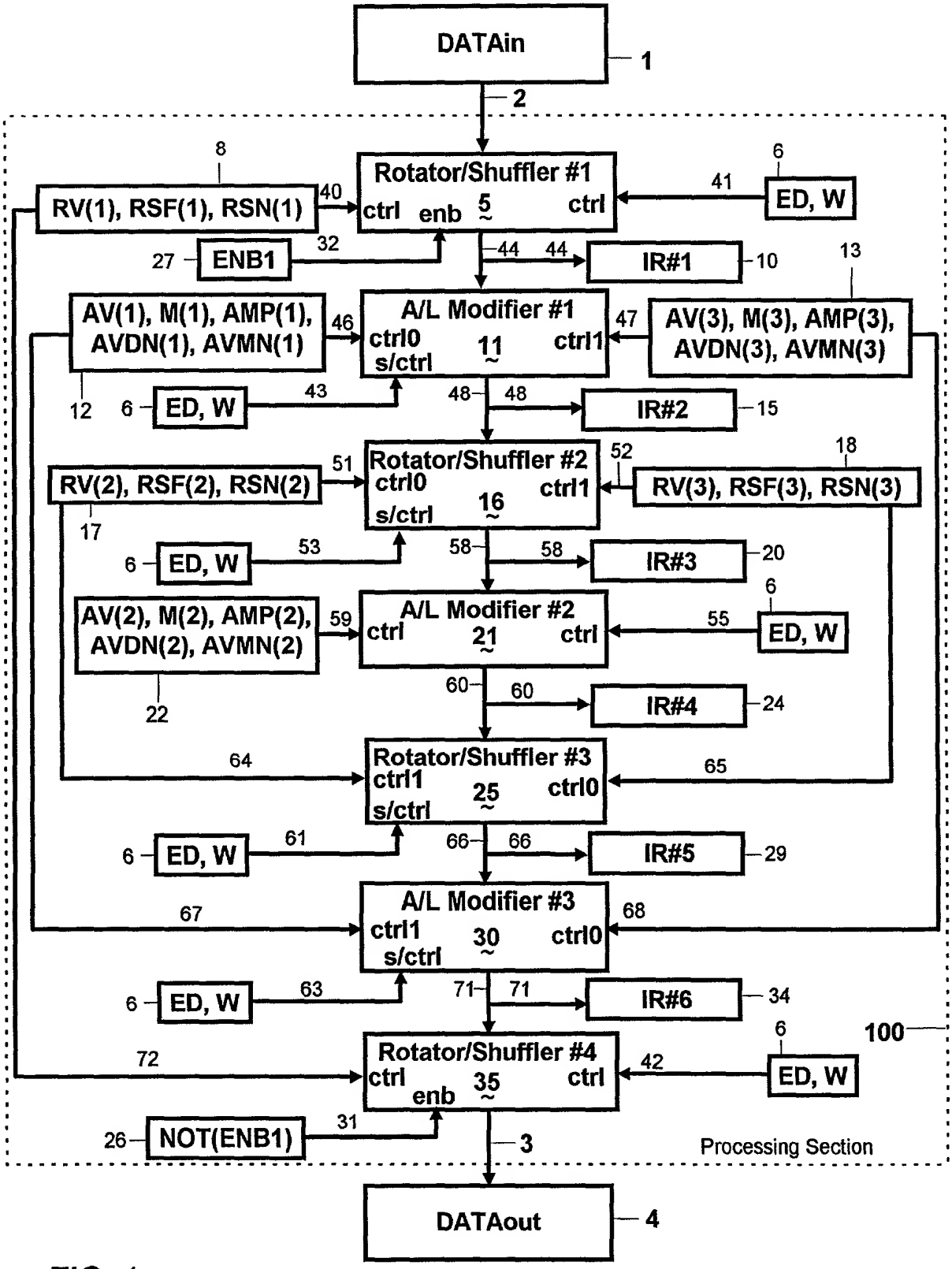

FIG. 1

VARIABLE WIDTH BLOCK CIPHER

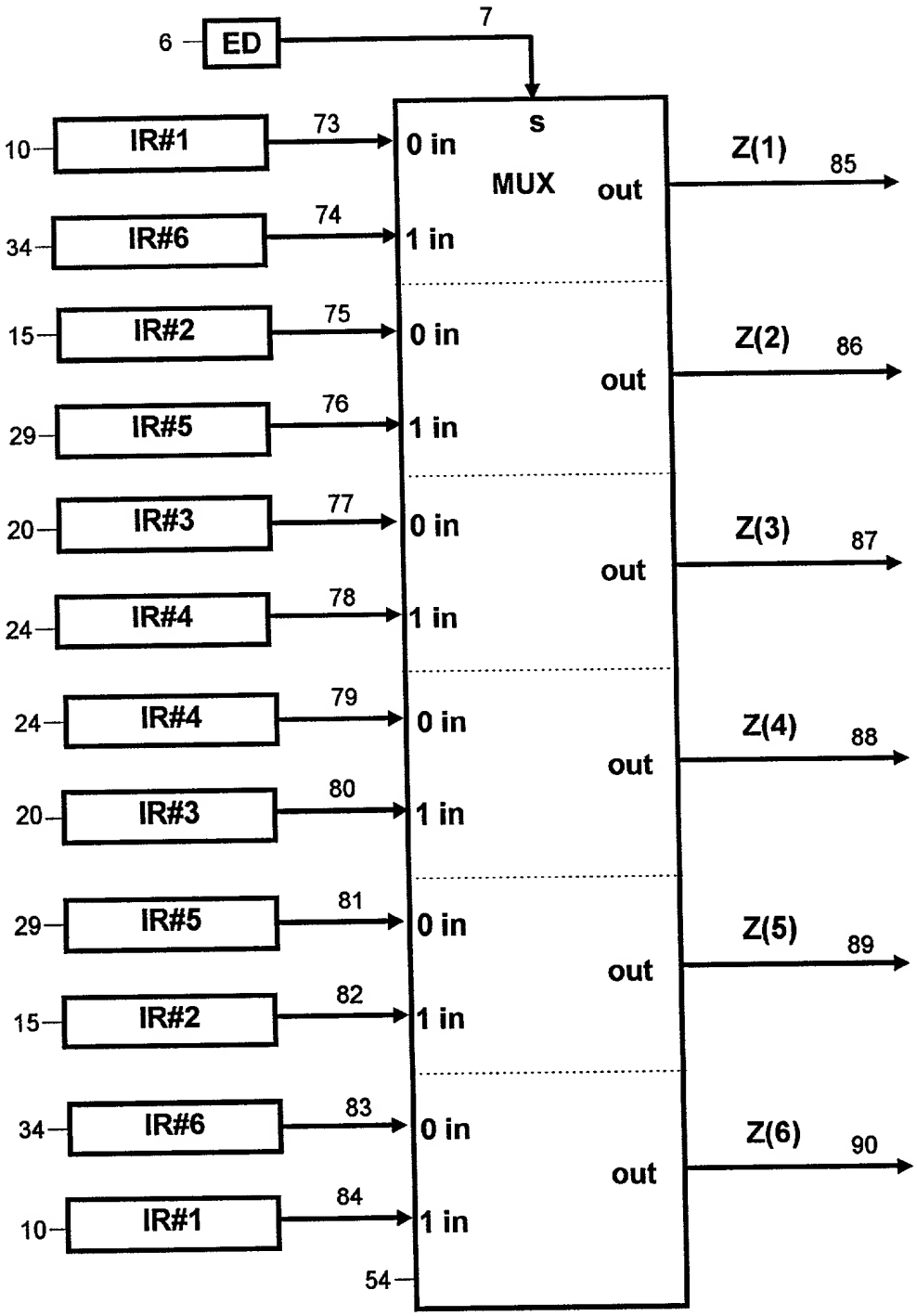

FIG. 1A

ELEMENTS OF A SOURCE POINTER

Change Enable Flag	Address Mode	Byte Pointer	Source Number
0=not changeable 1=change allowed	(see below)	P (or NP), relative byte pointer	RN, index into Source Dispatch Table (SDT)

ADDRESS MODES

Addressing Mode	Description
Fixed=0	Source and Pointer within source are constant
Jump=1	Both Source and Pointer values are changed by a local counter associated with a variable
Local=2	Source is constant, but pointer is incremented within a specified source
General=3	Pointer is incremented and at the end of one source, the next enabled source is used with the pointer reset to the beginning of the new source

SOURCE DISPATCH TABLE

Table Index, RN (Relative Source Number)	Table Entry, SN Bytes Sources Number
1	6
2	1
3	2
4	3
	0 (table terminator)

TNES = Total Number of Enabled Sources, (number of non zero entries in the Source Dispatch Table.

BYTES SOURCES

SN	Name	Description	Enabled	SL Length	CFlag
1	M(1)	first masking array	Always	ML(1)	
2	M(2)	second masking array	Always	ML(2)	
3	M(3)	third masking array	Always	ML(3)	
4	C(1)	first element of cross product of A x B	YES	x-2	
5	C(2)	second element of cross product of A x B	YES	x-2	
6	C(3)	third element of cross product of A x B	YES	x-2	
7	CD(1)	first element of cross product of A X D	YES	x-2	
8	CD(2)	second element of cross product of A X D	YES	x-2	
9	CD(3)	third element of cross product of A X D	YES	x-2	
10	CD(4)	first element of cross product of B X D	YES	x-2	
11	CD(5)	second element of cross product of B X D	YES	x-2	
12	CD(6)	third element of cross product of B X D	YES	x-2	
13	U(1)	first user defined calculation	YES	x	
14	U(2)	second user defined calculation	YES	x	
15	U(3)	third user defined calculation	YES	x	
16	U(4)	fourth user defined calculation	YES	x	
-	D(1)	first user defined array	No	N/A	
-	D(2)	second user defined array	No	N/A	
-	D(3)	third user defined array	No	N/A	

FIG. 2

Variables, Counters, Sources and Pointers

Name and Range	Description
MC(1-3)	Mask update counter
ML(1-3)	Mask lengths
MAV(1-3)	Method for updating Masks (1-3)
MRV(1-3)	Rotate value for Masks(1-3) when updating
MRSF(1-3)	Rotate/Shuffle indicator Flag when updating
MVARP(1-3)	Pointer and source for rotate/shuffle operation
Moff1(1-3)	Shuffle offset#1 value for updating Masks(1-3)
Moff2(1-3)	Shuffle offset#2 value for updating Masks(1-3)
MUP(1-3)	Pointer and source for mask update value
MURV(1-3)	Rotate Value for update item
MURSF(1-3)	Rotate/Shuffle flag for update item
MUVARP(1-3)	Pointer and source for rotate/shuffle operations
MUoff1(1-3)	Shuffle offset #1 for update value
MUoff2(103)	Shuffle offset #2 for update value
AC(1-3)	Counters for A/L Variable
ACP(1-3)	Pointer and source for updating AC(1-3)
AV(1-3)	A/L method variables
AMP(1-3)	Pointer into mask for retrieving mask values for A/L operations
AVDN(1-3)	Number base for converting data byes to digits
AVMN(1-3)	Number base for converting mask bytes to digits
AVP(1-3)	Pointer and source for updating AV(1-3)
RC(1-3)	Rotate/Shuffle counters
RCP(1-3)	Rotate/Shuffle counter update pointer
RENB(1)	Rotate Shuffle enable
RV(1-3)	Rotate values
RSN(1-3)	Number base for converting data bytes to digits
RSF(1-3)	Rotate/Shuffle selection flag
RVARP(1-3)	Pointer and source for for rotate/shuffle operation
RSoff1(1-3)	Offset#1 for R/S shuffle operation
RSoff2(1-3)	Offset#2 for R/S shuffle operation
MastC	Master counter
MastP	Pointer and source for updating Master Counter
MASTERMax	Maximum value for an Master Counter entry
ACMax	Maximum value for an AC entry
RCMax	Maximum value for an RC entry
MCMax	Maximum value for an MC entry
MSZ(1-3)	Maximum size for Masking Arrays(1-3)
W	Size of the Cipher Block in Bytes
ED	Encrypt/Decrypt Flag

FIG. 3

Initialization

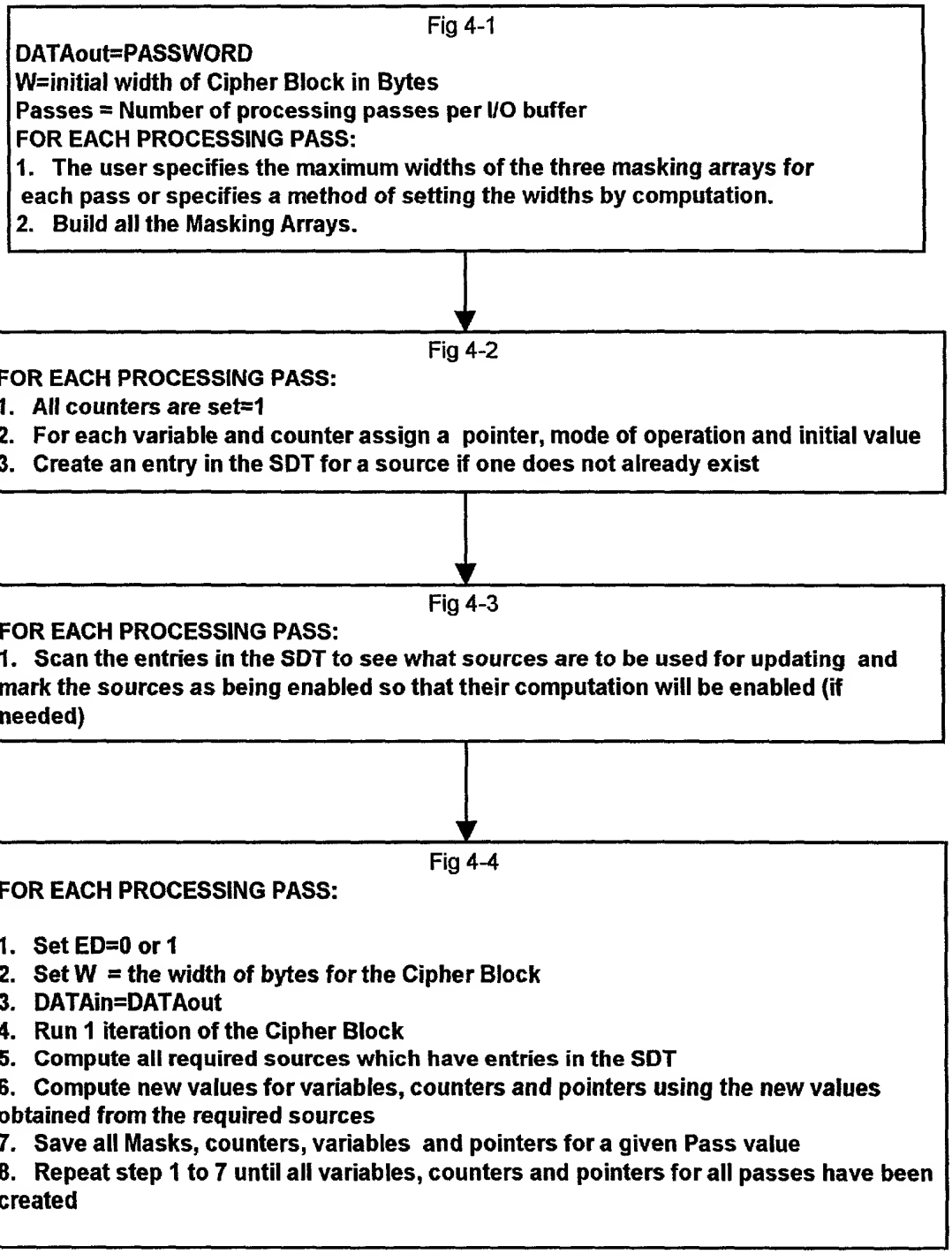

FIG. 4

Processing A Data File

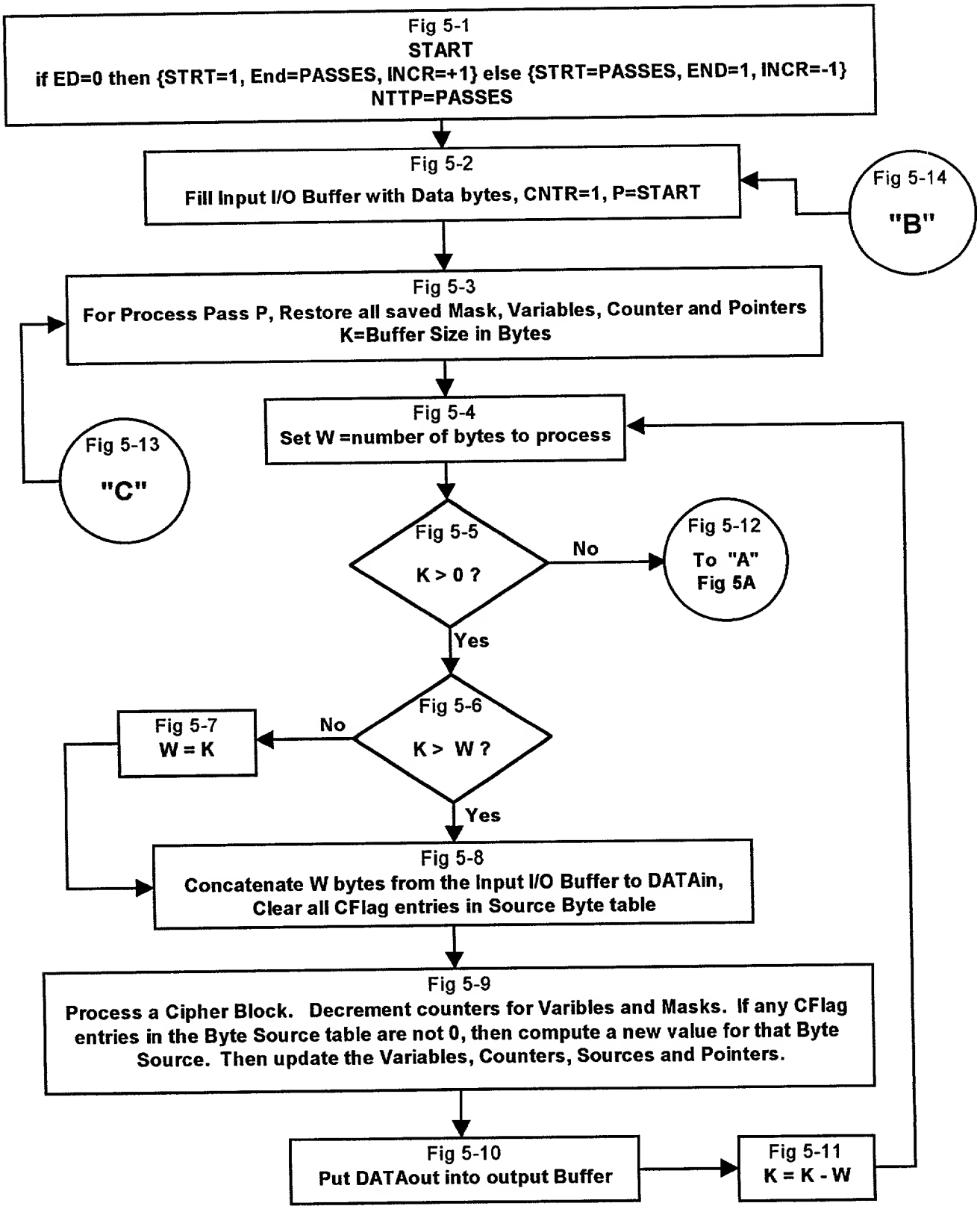

FIG. 5

PROCESSING A DATA FILE

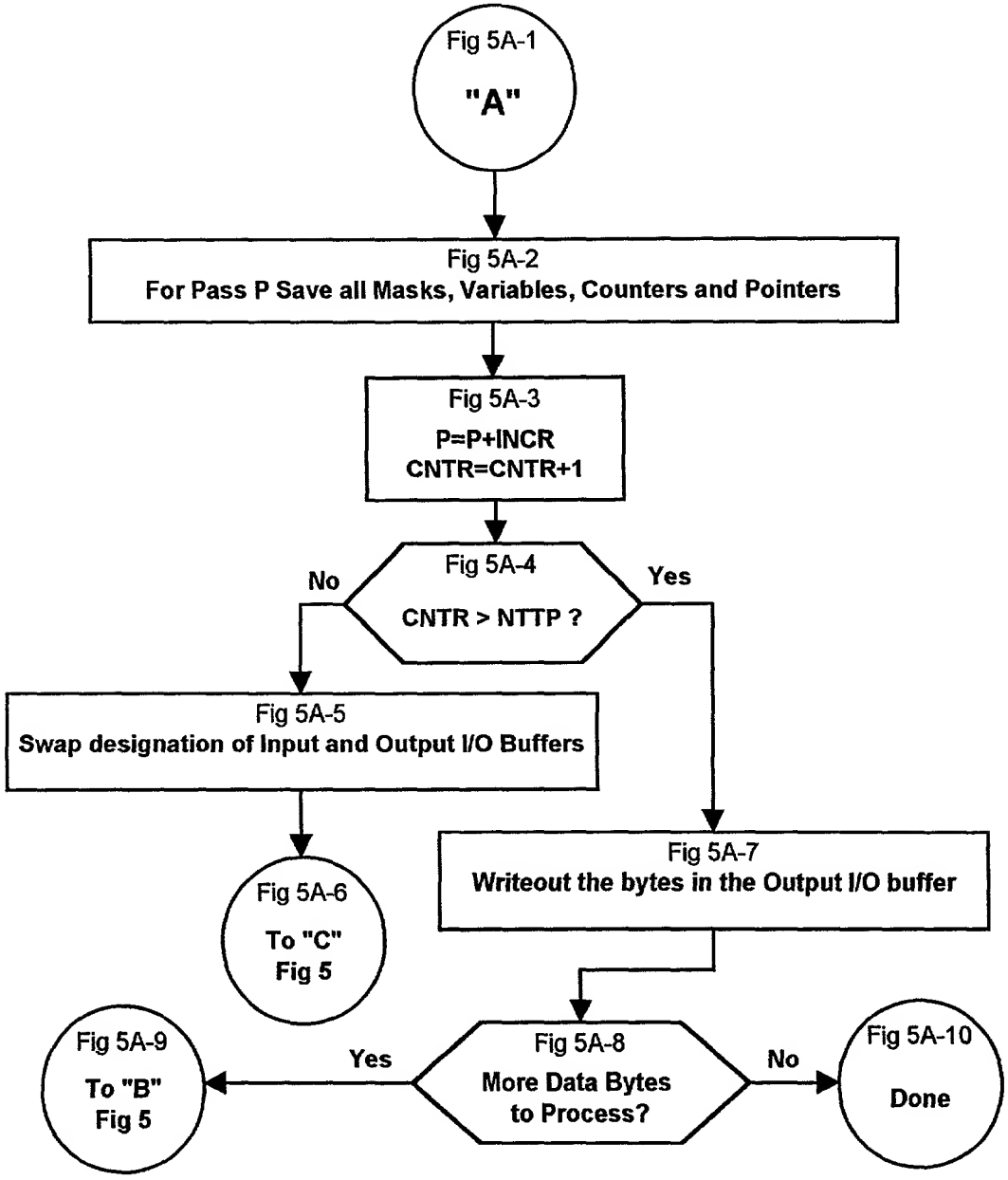

FIG. 5A

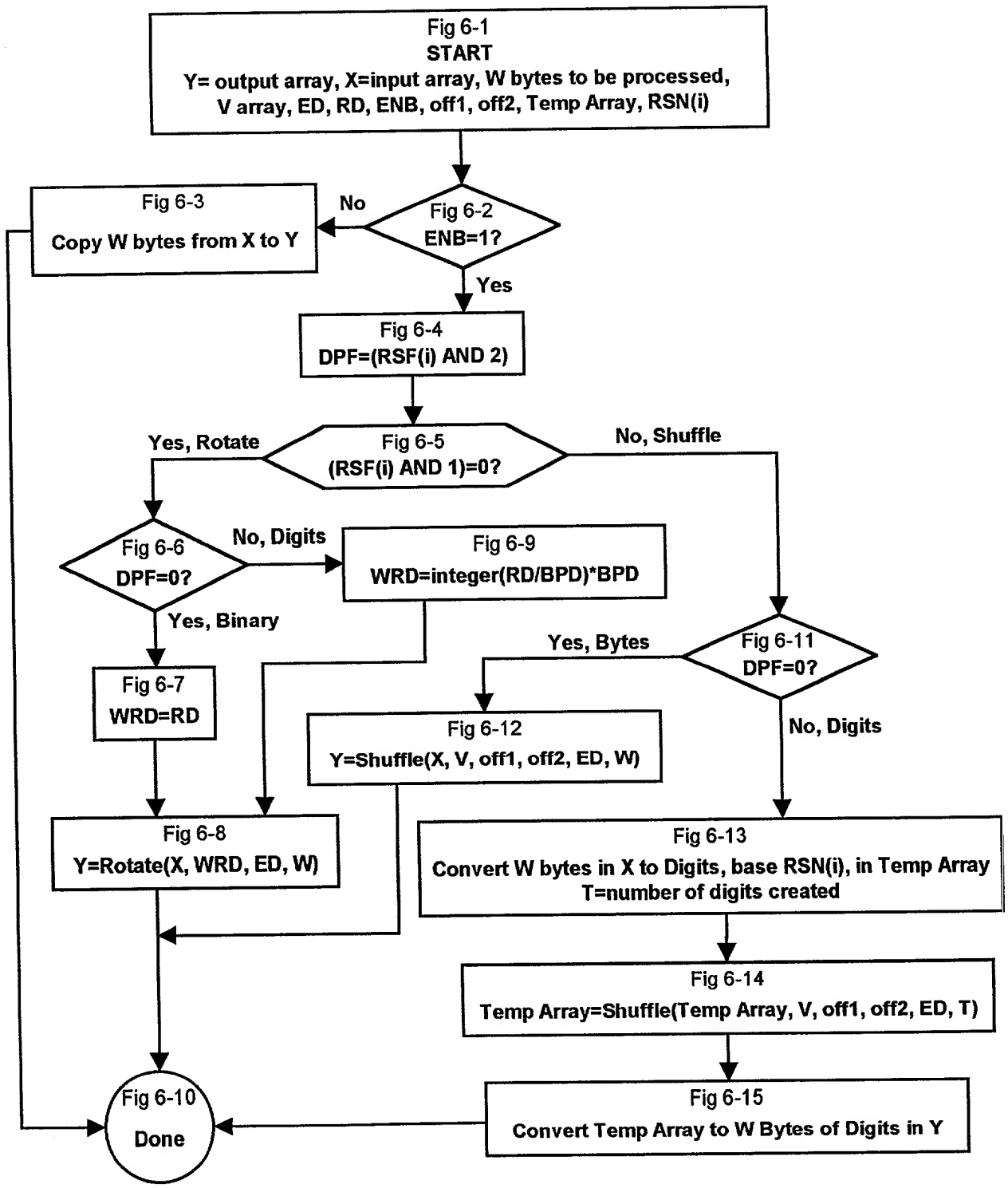
Flowchart - Rotate/Shuffle Operation
Fig 6-1
START
Y= output array, X=input array, W bytes to be processed,
V array, ED, RD, ENB, off1, off2, Temp Array, RSN(i)
Fig 6-2
ENB=1?
No
Fig 6-3
Copy W bytes from X to Y
Yes
Fig 6-4
DPF=(RSF(i) AND 2)
Fig 6-5
(RSF(i) AND 1)=0?
Yes, Rotate
No, Shuffle
Fig 6-6
DPF=0?
No, Digits
Fig 6-9
WRD=integer(RD/BPD)*BPD
Yes, Binary
Fig 6-7
WRD=RD
Fig 6-11
DPF=0?
Yes, Bytes
No, Digits
Fig 6-12
Y=Shuffle(X, V, off1, off2, ED, W)
Fig 6-8
Y=Rotate(X, WRD, ED, W)
Fig 6-13
Convert W bytes in X to Digits, base RSN(i), in Temp Array
T=number of digits created
Fig 6-14
Temp Array=Shuffle(Temp Array, V, off1, off2, ED, T)
Fig 6-10
Done
Fig 6-15
Convert Temp Array to W Bytes of Digits in Y

FIG. 6

Flowchart of Shuffle Operation

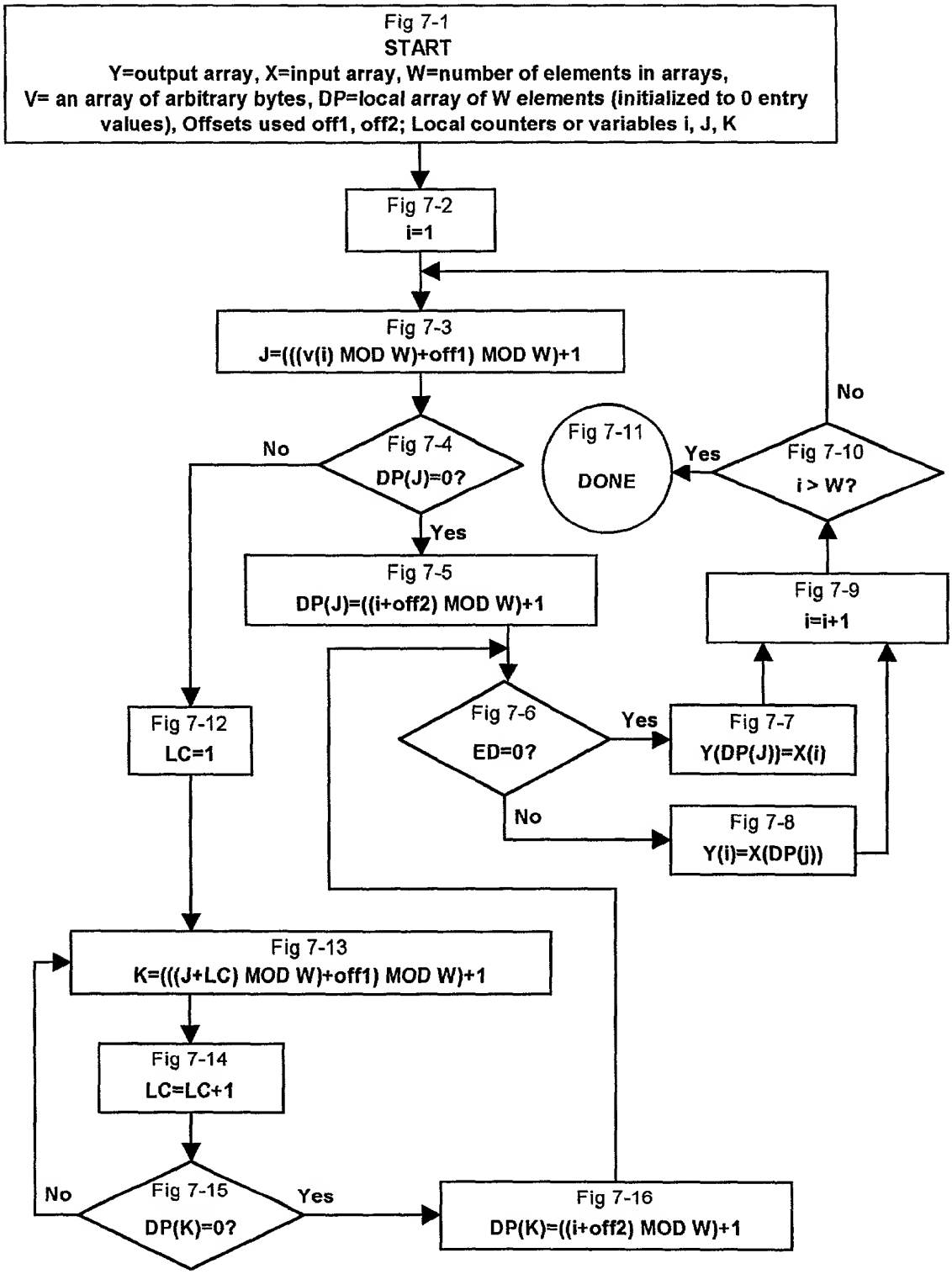

FIG. 7

Flowchart of Multibyte Binary Rotate

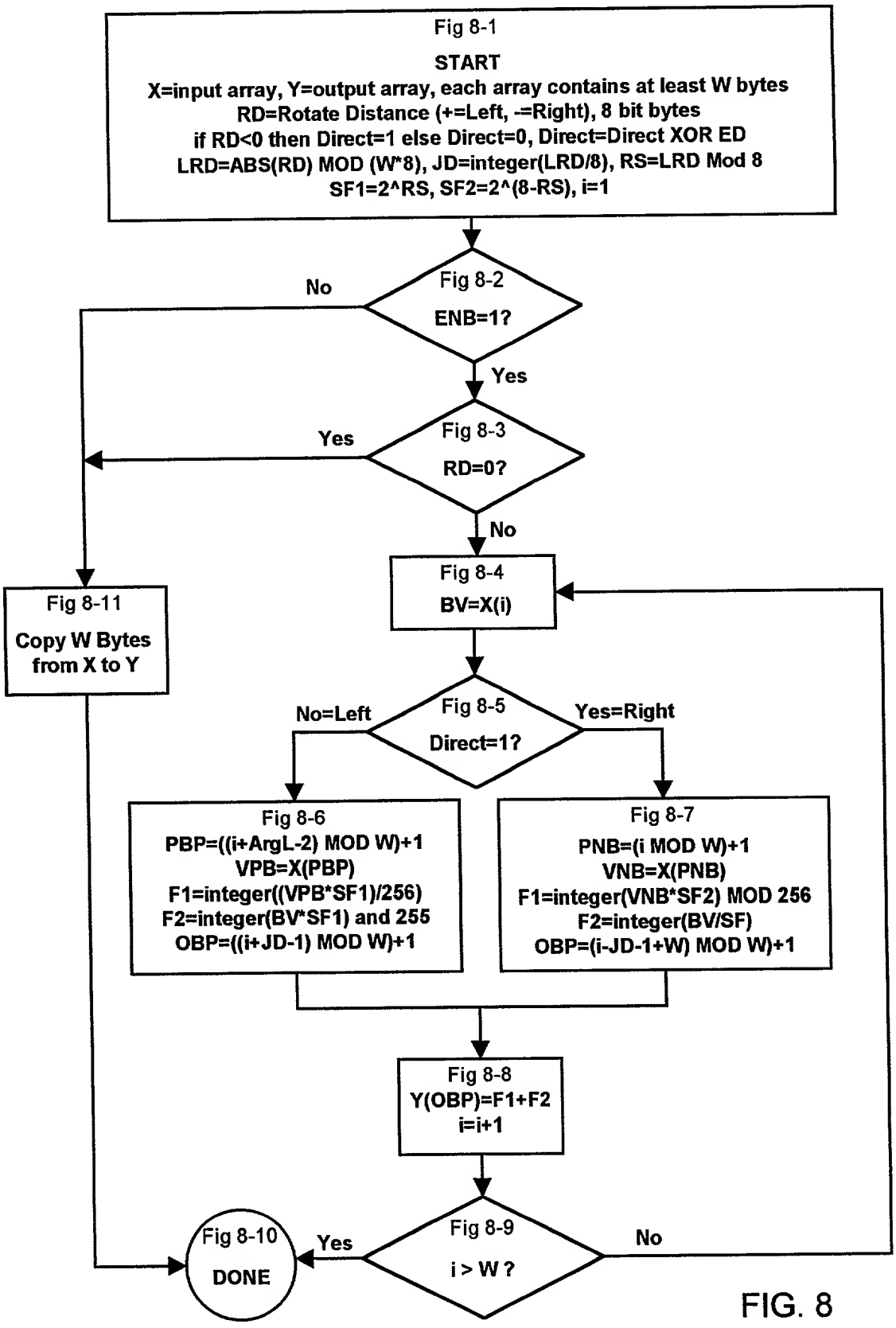

FIG. 8

ARITHMETIC/LOGIC OPERATIONS

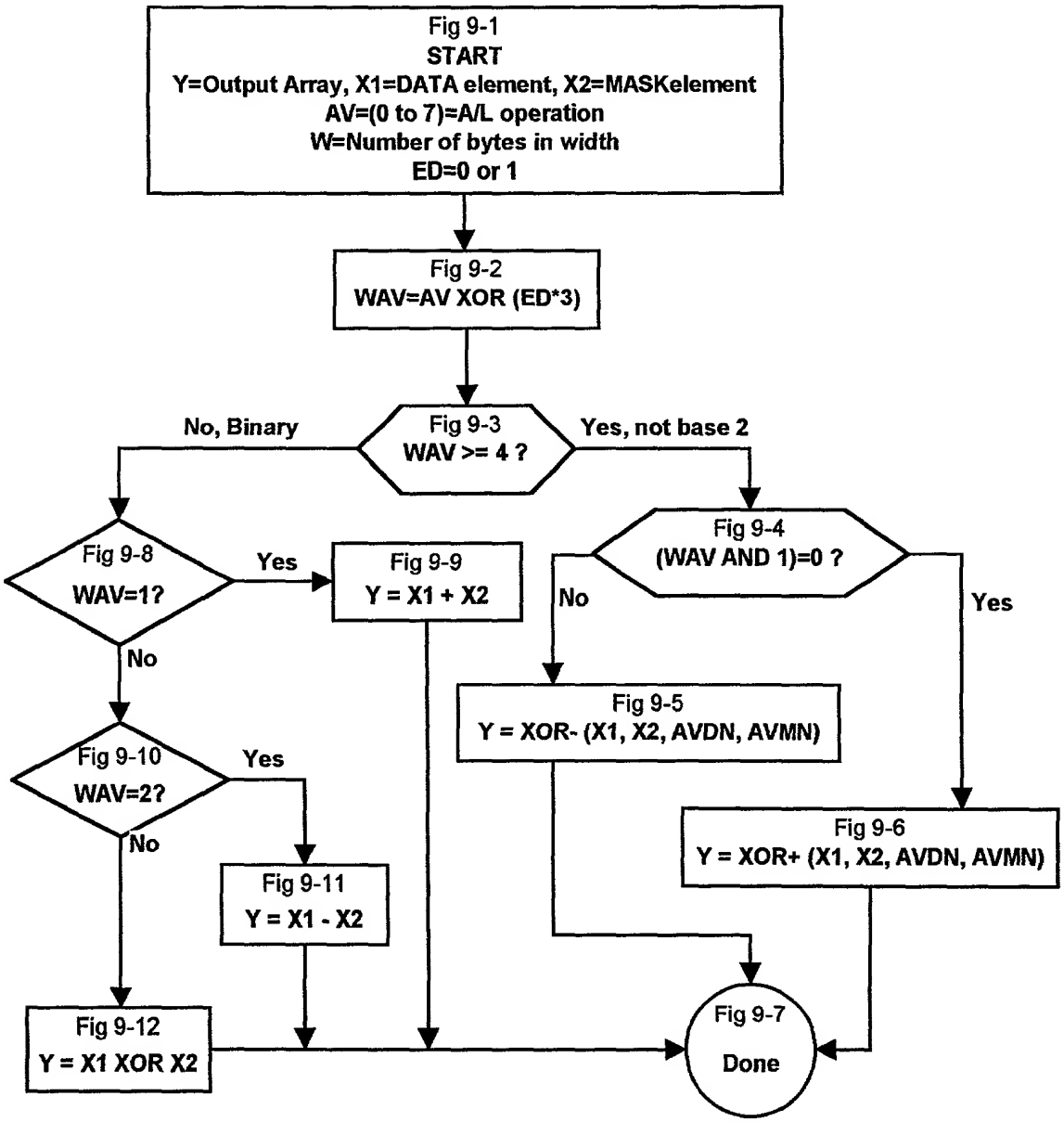

FIG. 9

FLOWCHART - UPDATING VALUE, SOURCE AND POINTER

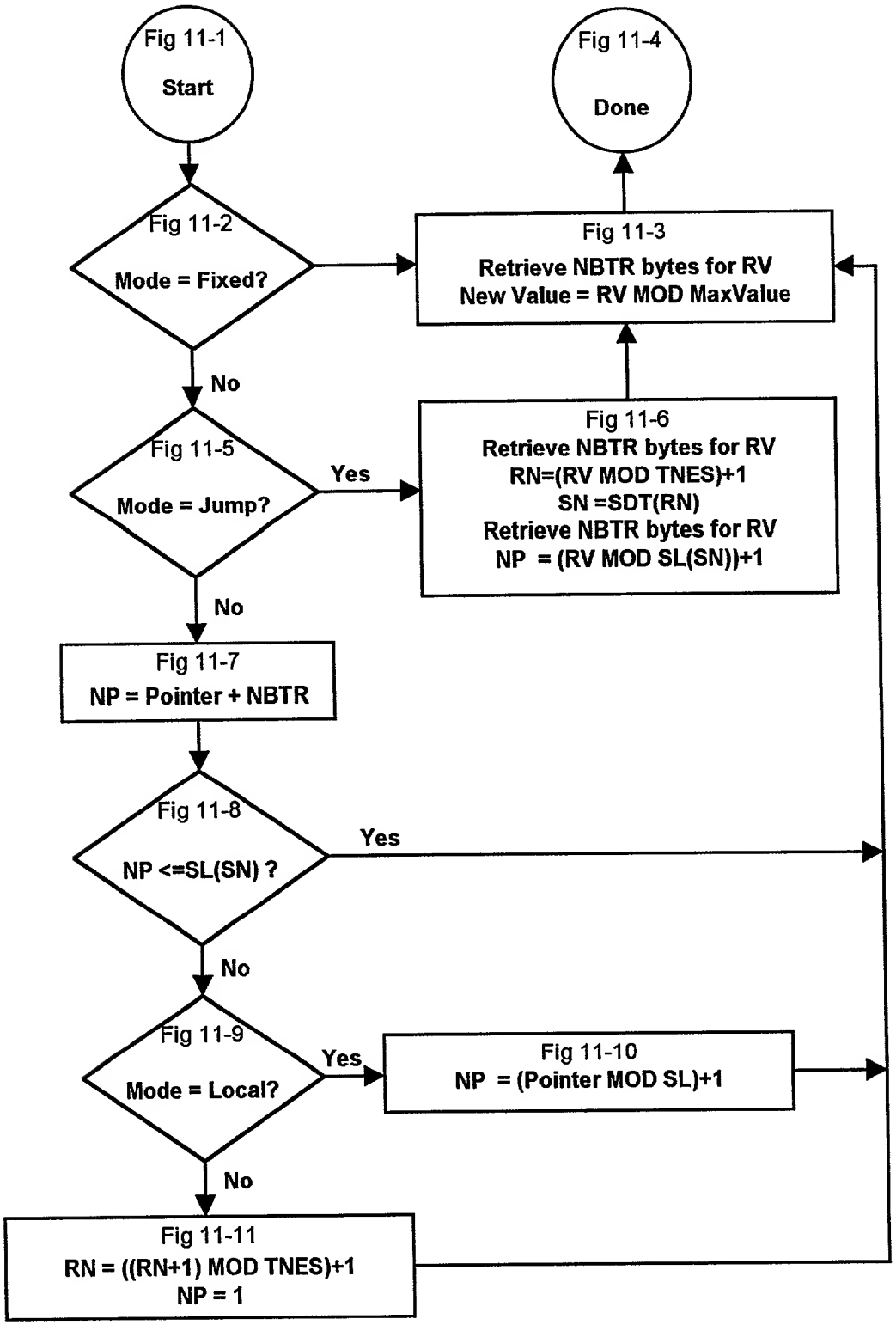

FIG. 11

RETRIEVING A VALUE

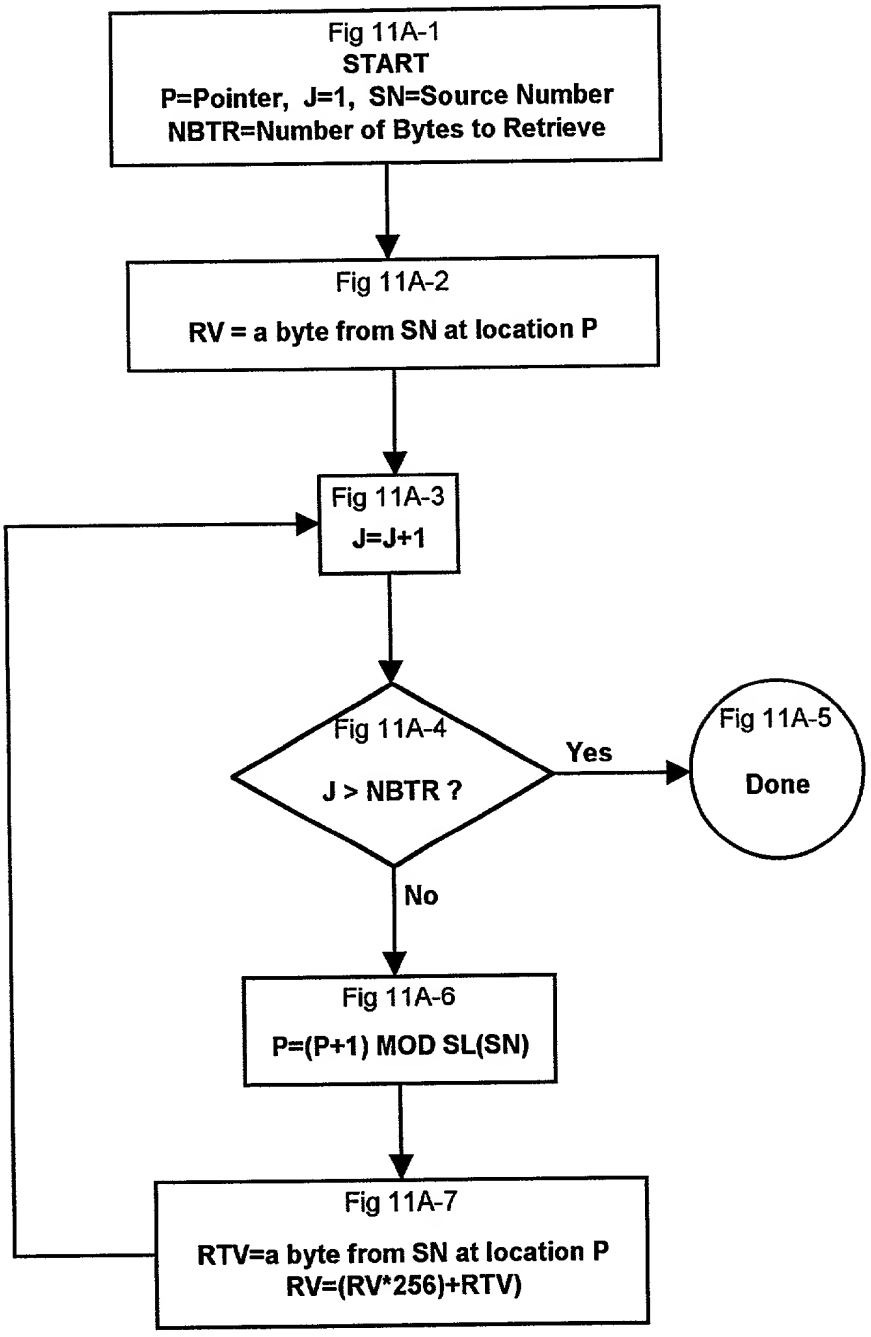

FIG. 11A

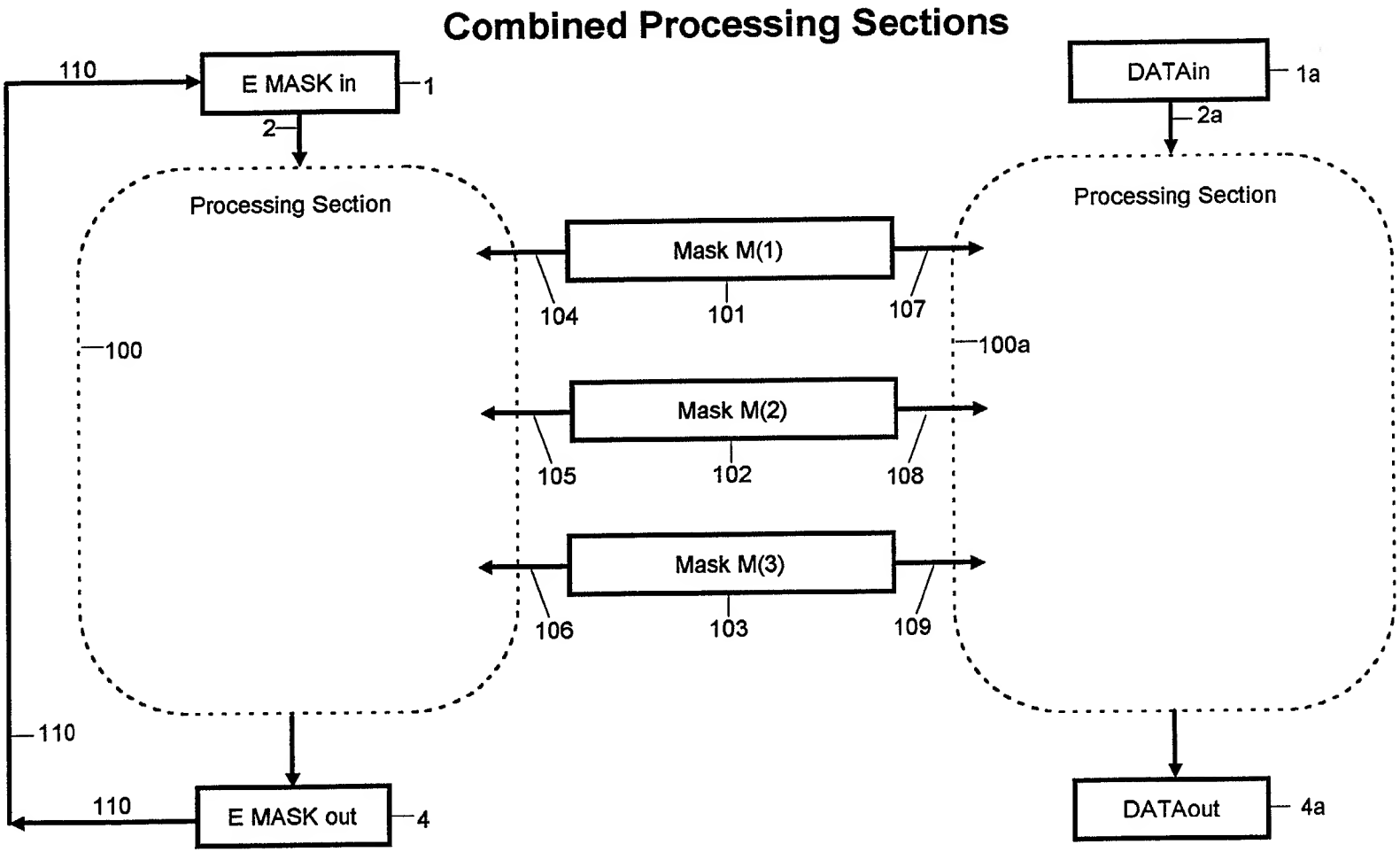
Combined Processing Sections
110
E MASK in
1
2
Processing Section
100
DATAin
1a
2a
Processing Section
100a
Mask M(1)
104
101
107
Mask M(2)
105
102
108
Mask M(3)
106
103
109
110
110
E MASK out
4
DATAout
4a

FIG. 12